نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید

نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید

سال اول شماره اول بهار و تابستان ۱۳۹۲، صفحه ۵۷-۷۱
www.ier.basu.ac.ir

ISSN: 2345-2250

توسعه روش جستجوی هماهنگی در حل مسائل بهینه‌سازی: مطالعه موردی در زمان‌بندی تولید ماشین‌های موازی

ایمان رستگار[1*]، راشد صحرائیان[†]

[*] کارشناس‌ارشد مهندسی صنایع، دانشگاه شاهد، تهران

[†] استادیار و عضو هیأت علمی گروه مهندسی صنایع، دانشگاه شاهد، تهران

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۱۳۹۲/۳/۱
پذیرش ۱۳۹۲/۶/۱۳

کلمات کلیدی:
بهینه‌سازی پیوسته،
زمان‌بندی،
روش‌های فراابتکاری تکاملی،
الگوریتم جستجوی هماهنگی،

خلاصه

الگوریتم جستجوی هماهنگی یک روش فراابتکاری جدید تکاملی بر مبنای فرایند موزیک است که با نسلی از بردارهای حل در قالب حافظه الگوریتم شروع به جستجوی فضای حل مسأله می‌کند و بر اساس رویکرد احتمالی به سمت فضاهای بهینه حرکت می‌کند. در این مقاله، ابتدا شرح کامل الگوریتم جستجوی هماهنگی ارائه می‌شود و سپس با توجه به معایب و مزایای این الگوریتم، ساختار جدیدی در مراحل الگوریتم ایجاد می‌شود. به طور مثال به منظور تعادل تنوع در ابتدای تکرارهای الگوریتم و شدت‌بخشی در انتهای تکرارهای الگوریتم، از روش‌های تنظیم پویای عملگرها و پارامترها استفاده می‌شود. همچنین، پارامترهای اولیه این الگوریتم با روش طراحی آزمایش‌های تاگوچی تنظیم می‌شود. جهت بررسی کارایی و بررسی عملکرد الگوریتم توسعه داده شده، روش پیشنهادی با دیگر روش‌های جستجوی هماهنگی شامل جستجوی هماهنگی بهبود داده شده و جستجوی بهترین کلی، جهت حل ۹ تابع هدف ریاضی پیوسته مقایسه شده است. نتایج حاصل شده، عملکرد مناسب روش جستجوی هماهنگی پیشنهادی را در اکثر نمونه‌ها نشان می‌دهد. همچنین، به‌منظور بررسی روش پیشنهادی در مسائل تولیدی، از این روش در حل مسأله زمان‌بندی ماشین‌های موازی غیریکسان نیز استفاده شده است.

۱- مقدمه

بهینه‌سازی در علوم ریاضی و کامپیوتر فرایند انتخاب یا پیدا کردن بهترین عضو در مجموعه‌ای از گزینه‌های موجود است. هر فرایندی پتانسیل بهینه شدن دارد و مسائل پیچیده می‌توانند در زمینه‌های علوم مهندسی، اقتصادی و تجاری به‌صورت مسائل بهینه‌سازی مدل‌سازی شوند. هدف از مدل‌سازی مسائل بهینه‌سازی حداقل کردن زمان، هزینه و ریسک یا حداکثر کردن سود، کیفیت و اثربخشی است. بعضی از مسائل بهینه‌سازی پیچیده هستند و به‌دست آوردن جواب‌های بهینه در زمان معقول با روش حل دقیق مانند روش‌های برنامه‌ریزی پویا و شاخه و کران مشکل است. از این رو توسعه روش‌های حل در این نوع مسائل که بتوانند در زمان معقول جواب‌های بهینه یا نزدیک به بهینه به‌دست آورند، از نظر اقتصادی به صرفه‌تر است. در سالیان اخیر محققان در اکثر مسائل پیچیده بهینه‌سازی با پیاده‌سازی روش‌های فراابتکاری به نتایج مناسبی دست یافته‌اند.

الگوریتم جستجوی هماهنگی[2] (HS) یکی از ساده‌ترین و جدیدترین روش‌های فراابتکاری است که در فرایند جستجوی جواب شدنی بهینه در مسائل بهینه‌سازی، از فرایند نواختن همزمان گروه ارکستر موزیک الهام گرفته شده است. به‌عبارت دیگر، میان پیدا

[1] نویسنده مسئول.
تلفن: ۰۲۱-۵۱۲۱۲۰۹۲ پست الکترونیک: sahraeian@shahed.ac.ir

2. Harmony Search

کردن یک حل بهینه در مسأله پیچیده و فرایند اجرای موزیک تشابه وجود دارد.این روش حل را ابتدا گیم در سال ۲۰۰۱ میلادی ارائه کرد [۱]. مطابق با منطق این روش فراابتکاری، تلاش برای به‌دست آمدن هماهنگی در یک فرایند موزیک، مشابه پیدا کردن حل بهینه در مسائل بهینه‌سازی است. HS مانند ژنتیک، جزء الگوریتم‌های بهبود دهنده است. به عبارت دیگر، با نسلی از بردارهای حل شروع و برای ایجاد نسل‌های جدید از فرایند انتخاب استفاده می‌شود. اما بر خلاف الگوریتم ژنتیک (که در آن از دو کروموزم برای تولید کروموزوم یا بردار حل جدید استفاده می‌شود) در این روش از همه بردارهای حل موجود در حافظه برای تولید (Improvise) حل جدید استفاده می‌شود. از مزایای این الگوریتم، همگرایی سریع آن به دلیل ساختار مناسب آن است و از معایب آن افتادن در دام محلی به دلیل جستجو با تنوع کم در تکرارهای پایانی الگوریتم است که برای رفع آن از تکنیک فاز شروع دوباره و تغییر در قواعد الگوریتم به خصوص در تکرارهای پایانی استفاده می‌شود.

در این مقاله از مسأله‌ای کاربردی در زمینه مباحث تولیدی به‌منظور ارزیابی کارایی روش فراابتکاری پیشنهادی، استفاده می‌شود. یکی از مسائل مهم و پرکاربرد در زمینه بهینه‌سازی مهندسی، مسأله زمان‌بندی است. هدف از زمان‌بندی، تخصیص مجموعه‌ای از کارها به مجموعه‌ای از ماشین‌ها و تعیین اولویت انجام کارها به نحوی است که بهره‌وری منابع (مانند اپراتورها وماشین‌ها)حداکثر شود.

از مهمترین محیط‌های مسائل زمان‌بندی، محیط ماشین‌های موازی است که در آن، هر ایستگاه کاری از حداقل ۲ ماشین یکسان تشکیل شده است. اهمیت محیط ماشین‌های موازی از آنجاست که احتمال توقف برنامه‌ریزی تولید در آن کمتر است.به عبارت دیگر، در شرایطی که از یک ماشین در ایستگاه کاری استفاده نماییم، خرابی این ماشین موجب توقف خط تولید خواهد شد. با توجه به این اهمیت، ایستگاه‌های کاری با ماشین‌های موازی، در صنایع جدید مورد توجه اکثر محققان قرار گرفته است [۲].

ادامه ساختار این مقاله به این صورت است که در بخش دوم، یک مسأله تولیدی در زمینه مسائل زمان‌بندی ارائه می‌شود. در بخش سوم، پژوهش‌های محققان پیشین در زمینه توسعه روش‌های فراابتکاری (و به صورت دقیق‌تر، الگوریتم جستجوی هماهنگی) مرور می‌شود. در بخش چهارم، معایب و مزایای روش جستجوی هماهنگی معرفی و بر مبنای آن، روش جستجوی هماهنگی جدید پیشنهادی ارائه می‌گردد. در بخش پنجم پس از معرفی توابع ریاضی پیوسته، نتایج حاصل از آزمایش‌ها، با سایر روش‌های جستجوی هماهنگی مقایسه و ارزیابی می‌شوند. در بخش ششم، مسأله زمان‌بندی ماشین‌های موازی غیریکسان با روش پیشنهادی حل می‌شود و نتایج آن با نتایج روش جستجوی هماهنگی ساده مقایسه و ارزیابی می‌شود. در بخش آخر نیز، نتیجه‌گیری و زمینه‌های تحقیق آتی این مقاله توضیح داده می‌شود.

۲- تعریف مسأله

در مسائل زمان‌بندی، هزینه‌های تولید از مهمترین هزینه‌های سیستم است که با کاهش زمان کل تولید یعنی حداکثر زمان تکمیل در یک سیستم زمان‌بندی به دست می‌آید. فرض می‌شود که n کار وجود دارد که باید روی m ماشین موازی با سرعت‌های متفاوت، پردازش شوند. فرض می‌شود ماشین اول دارای سرعت v_1=۱است و داریم:

$v_1 < v_2 < \ldots < v_m$ **رابطه ۱**

در نتیجه ماشین m ام کارها را با سرعت بالاتری پردازش می‌کند و عملیات پردازش آن قطعه سریع‌تر به پایان می‌رسد. هدف از زمان‌بندی این است که کارها به نحوی به ماشین‌های موازی تخصیص یابند که زمان تکمیل آخرین کار روی آخرین ماشین (Makespan) حداقل شود. می‌توان فرض کرد که شرایط و ویژگی‌های زیر در یک مسأله زمان‌بندی، برقرار است:

- زمان پردازش کارها معلوم و قطعی است.
- حق انقطاع در هنگام پردازش کارها وجود ندارد.
- هر ماشین در هر لحظه می‌تواند تنها یک قطعه را پردازش نماید.
- هر کار تنها روی یکی از ماشین‌های موازی پردازش می‌شود و از ایستگاه خارج می‌شود.
- همه کارها و ماشین‌ها در لحظه شروع دوره زمان‌بندی در دسترس هستند.

۳- مرور ادبیات

در دهه‌های اخیر برای حل مسائل بهینه‌سازی، روش‌های مختلفی توسعه داده شده است. این الگوریتم‌ها به دو دسته قطعی و احتمالی طبقه بندی می‌شوند [۳]. الگوریتم‌های قطعی روش‌های جستجوی محلی بر پایه گرادیان هستند که به اطلاعات اساسی حرکتی برای پیدا کردن یک جواب شدنی نیاز دارند[۴]. در مسائلی که فضای حل غیر محدب باشد، پیدا کردن جواب بهینه سراسری با استفاده از این الگوریتم‌ها کار آسانی نیست. به عبارتی، بعضی از مسائل که غیر محدب هستند شامل چندین بهینه محلی در فضای حل‌شان هستند. در چنین مسائلی کیفیت جواب‌های نهایی به مقادیر اولیه معینی نیاز دارد. به همین دلیل محققین، در پژوهش‌ها به استفاده از الگوریتم‌های بهینه سازی احتمالی روی آورده‌اند. این الگوریتم‌ها از پدیده‌های طبیعی الهام گرفته‌اند و به اطلاعات حرکتی برای پیدا کردن جواب نیاز ندارند. مهم‌ترین الگوریتم‌های بهینه‌سازی احتمالی شامل الگوریتم ژنتیک، جستجوی ممنوع، بهینه‌سازی ذرات گروهی، شبیه سازی تبرید، کلونی مورچگان و جستجوی هماهنگی است.

در سالیان اخیر الگوریتم‌های ترکیبی محلی سراسری برای حل مسائل غیر محدب با موفقیت اجرا شده‌اند [۵]. این الگوریتم‌ها جنبه‌های جستجوی محلی و جستجوی سراسری را در ساختارشان برای جستجوی مؤثر ترکیب می‌کنند. در این الگوریتم‌ها فرایند

جستجوی سراسری با چندین نقطه شروع و جستجوی فضای حل از این نقاط شروع می‌شود. سپس الگوریتم‌های جستجوی محلی، حل بهینه را در این فضاها پیدا می‌کنند. در حوزه الگوریتم‌های بهینه‌سازی ترکیبی، پژوهش‌های فراوانی انجام شده است. در یک مورد، روش ژنتیک و روش تندترین شیب، ادغام شد که منجر به طراحی روش ژنتیک ترکیبی شد. در این پژوهش، عملکرد روش پیشنهادی با روش ژنتیک ساده برای حل دو تابع هدف چند مدله آزمایش شد. نتایج نشان داد که روش ژنتیک ترکیبی، ۷۵٪ زمان کمتری نسبت به روش ژنتیک ساده برای به دست آوردن جواب بهینه نیاز دارد [۶]. در تحقیق دیگری، یک مدل ترکیبی از الگوریتم بهینه سازی ذرات گروهی و روش سیمپلکس Nelder-mead توسعه داده شد[۷]. آن‌ها عملکرد مدل ترکیبی را روی ۱۳ تابع هدف بهینه‌سازی پیوسته آزمایش کردند. نتایج نشان داد که مدل ترکیبی، در جستجوی جواب‌های بهینه موثرتر و کاراتر است. محققین سپس، روش شبیه سازی تبرید و روش برنامه‌ریزی درجه دوم متوالی شدنی را ترکیب کردند [۸]. مدل ترکیبی آن‌ها شامل ۲ رویکرد حل است. نتایج ۳۲ مسأله پیوسته بهینه سازی نشان داد که این مدل ترکیبی جواب‌های بهتری از سایر روش‌های حل پیدا می‌کند. در ادامه، یک مدل ترکیبی از روش‌های جستجوی ممنوعه و برنامه‌ریزی درجه دوم متوالی برای حل مسأله بهینه‌سازی ارائه شد[۹]. این مدل شامل دو رویکرد حلقه داخلی و حلقه خارجی برای جستجوی فضای حل شدنی است. برنامه‌ریزی درجه دوم متوالی در حلقه داخلی به عنوان یک روش جستجوی محلی استفاده می‌شود و جستجوی ممنوعه در حلقه خارجی به عنوان جستجوگر سراسری استفاده می‌شود.

۳-۱- الگوریتم جستجوی هماهنگی

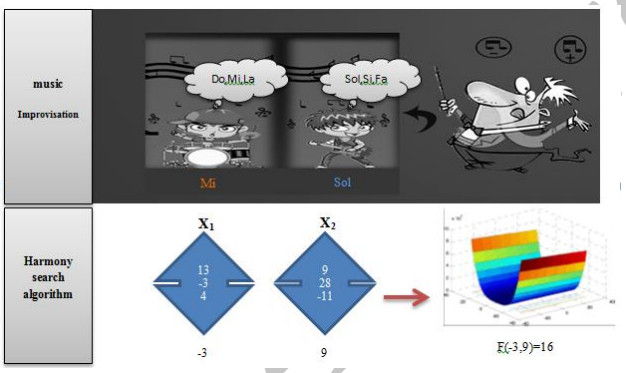
الگوریتم HS به دلیل کاربردی بودن برای مسائل بهینه‌سازی گسسته و پیوسته، محاسبات ریاضیاتی کم، مفهوم ساده، پارامترهای کم و اجرای آسان به یکی از پرکاربردترین الگوریتم‌های بهینه‌سازی در سال‌های اخیر در مسائل مختلف تبدیل شده است. این الگوریتم در مقایسه با سایر روش‌های فراابتکاری الزامات ریاضیاتی کمتری دارد و می‌توان آن را در مسائل مختلف مهندسی با تغییر در پارامترها و عملگرها تطبیق نمود[۱۰ و ۱۱].

این الگوریتم به طور صعودی در سالیان اخیر توجه زیادی به خود معطوف کرده به طوری که تاکنون در مسائل بهینه‌سازی عملی نظیر بهینه‌سازی ساختاری، تخمین پارامترهای مدل غیر خطی Muskingum، طراحی بهینه شبکه‌های توزیع آب، مسیریابی، طراحی اسکلت‌های فلزی، مدل‌های انتقال انرژی، زمان‌بندی و غیره به کار رفته است.در بعضی از تحقیقات نشان داده شده است که HS در به‌دست آوردن جواب‌های مناسب، در زمان زودتری نسبت به روش ژنتیک عمل می‌کند[۴ و ۱۰].

از مزیت‌های دیگر این روش نسبت به روش ژنتیک این است که برای ایجاد حل جدید برخلاف روش ژنتیک که از دو بردار حل در نسل استفاده می‌کند، این الگوریتم از همه حل‌های موجود در حافظه‌اش استفاده می‌کند. این ویژگی انعطاف الگوریتم را در جستجوی فضاهای بهتر حل افزایش می‌دهد.

از ویژگی‌های دیگر الگوریتم جستجوی هماهنگی این است که در مدت زمان مناسبی فضاهای حل با محدوده عملکرد بهتری را شناسایی می‌کند. این ویژگی در صورتی که مسأله مورد مطالعه از بهینگی محلی (Local Optimum) برخوردار باشددچار مشکل می‌شود و در بهینگی محلی متوقف شده و نمی‌تواند به بهینگی سراسری (Global Optimum) برسد. دلیل این مشکل عدم کارایی مناسب الگوریتم در اجرای جستجوی محلی در مسائل بهینه‌سازی گسسته است [۱۰]. به‌منظور رفع این مشکل و تطبیق روش حل با مسأله، محققان با تغییر در پارامترها و عملگرهای الگوریتم، انواع مختلفی از این الگوریتم ارائه کردند تا دقت حل و نرخ همگرایی افزایش یابد. *PAR* و *HMCR* دو پارامتری هستند که نقش اساسی در تنظیم صحیح بردارهای حل جدید دارند.

مطابق با شکل ۱، الگوریتم HS از فرایند بداهه سرایی موزیک گروه ارکستر پیروی می‌کند.هر نوازنده موزیک گام‌هایی از ابزارهای موسیقی خود را می‌نوازد تا شرایطی بهتر از هماهنگی در ارکستر به وجود آورد. هدف از این فرایند رسیدن به شرایطی است که هماهنگی کاملی از نواها ایجاد گردد. خروجی این هماهنگی کامل، صدای خوشایندی است که با استانداردهای زیبا مقایسه می‌گردد.



شکل ۱:تشابه الگوریتم جستجوی هماهنگی و بداهه سرایی موزیک

- **ساختار روش حل**

در این الگوریتم هر حل یک هارمونی نامیده می‌شود و با یک بردار Nبعدی نمایش داده می‌شود.این الگوریتم سه فاز اصلی دارد: ۱- نسل اولیه (مقداردهی اولیه) ۲- بهبود بردار هارمونی جدید (Improvisation) ۳- به‌روز کردن حافظه الگوریتم.

مطابق با فاز اول، یک نسل اولیه از بردارهای هارمونی به طور تصادفی ایجاد و در حافظه هارمونی[1](HM) ذخیره می‌شود. در فاز دوم یک بردار هارمونی جدید (حل جدید) با استفاده از قواعد در نظر گرفتن حافظه، تطبیق گام و دوباره نسل سازی تصادفی از حل‌های

1. Harmony Memory Size

موجود در حافظه الگوریتم ایجاد می‌گردد.

- **شرح روش حل**

گام‌های الگوریتم فوق عبارتند از:

گام اول: تعیین مقادیر اولیه پارامترهای الگوریتم

گام دوم: تعیین مقادیر ابتدایی حافظه هارمونی

گام سوم: تولید بردار حل جدید (Improvisation)

گام چهارم: به‌روزآوری حافظه هارمونی

گامپنجم : تست قاعده توقف

۱- ابتدا پارامترهای الگوریتم تعیین می‌گردد. پارامترهای الگوریتم HS شامل اندازه حافظه هارمونی(*HMS*)(تعداد بردارهای حل در حافظه)، نرخ در نظرگرفتن حافظه هارمونی[1](*HMCR*)، نرخ تطبیق گام[2](*PAR*)، فاصله پهنای باند[3](*BW*) و تعداد بهبودها[4](*NI*) (شرط توقف این روش حل) هستند.

۲- جمعیتی از بردارهای حل مطابق شکل ۲در ماتریس حافظه الگوریتم جستجوی هماهنگی شامل *HMS* بردار حل به صورت تصادفی ثبت می‌گردد. مؤلفه های هر بردار حل باید در محدوده آن قرار داشته باشد.

1	X_1^1	X_2^1	. . .	X_N^1	**f(x1)**
2	X_1^2	X_2^2	. . .	X_N^2	$f(x_2)$
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
HMS-1	X_1^{HMS-1}	X_2^{HMS-1}	. . .	X_N^{HMS-1}	$f(x_{HMS-1})$
HMS	X_1^{HMS}	X_2^{HMS}	. . .	X_N^{HMS}	$f(x_{HMS})$

شکل ۲: حافظه الگوریتم با HMS **بردار حل**

۳- در این مرحله بردار حل جدید تولید می‌شود. مطابق با شبه کد زیر به‌منظور تولید حل جدید ابتدا مؤلفه اول به دست می‌آید و این کار تا مؤلفه Nام بردار حل تکرار می‌شود. روند ایجاد هر مؤلفه مطابق با ۳ قاعده است. قاعده اول(Memory Consideration)، قاعده در نظرگرفتن حافظه است. در این قاعده اگر مقدار rand (مقداری تصادفی بین صفر و یک از تابع توزیع احتمال یکنواخت است) از *HMCR* کوچکتر باشد مقدار مؤلفه i-ام حل جدید از مؤلفه i-ام یکی از حل‌های موجود در حافظه به‌صورت تصادفی مقداردهی می‌شود. در غیر این صورت مطابق با قاعده سوم (Selection Random) مقدار مؤلفه به صورت تصادفی از محدوده مشخص آن مؤلفه مقداردهی می‌شود. از قاعده دوم الگوریتم (Adjustment Pitch) زمانی که قاعده اول اجرا شد استفاده می‌شود که مطابق با آن تغییری متناسب با مقدار *BW*در مقدار مؤلفه حل جدید درصورتی که مقدار rand از *PAR*کوچکتر باشد ایجاد می‌شود. پارامتر i در شکل ۳، مؤلفه i-ام یک حل (هارمونی) است. به طور جداگانه در یک حلقه و در هر تکرار برای کل مؤلفه‌های هر حل این کار تکرار می‌شود.

```
for (j=1 to n) do
  if (r1< HMCR) then
    Xnew(j) = Xa(j) where a ∈ (1,2,…,HMS)
    if (r2< PAR) then
      Xnew(j) = Xnew(j) ± r 3 × BW where r1,r2,r3 ∈ (0,1)
      endif
  else
      Xnew(j) = LBj + r × (UBj -LBj), where r ∈ (0,1)
  endif
endfor
```

شکل ۳: شبه کد تولید حل جدید در روش جستجوی هماهنگ [۱۲]

۴- اگر مقدار تابع هدف حاصل از بردار حل جدید از مقدار تابع هدف بدترین حل موجود در حافظه بهتر بود جایگزین آن در حافظه هارمونی می‌گردد(حافظه به روز می‌شود).

۵- فرایند بالا ادامه می‌یابد تا شرط توقف که تعداد بردارهای جدید ایجاد شده است (*NI*)، حاصل گردد.

شیوه نمایش جواب مطابق شکل ۴ با توجه به ماهیت پیوسته الگوریتم، نوع Random Key است. بین مقادیر حد پایین ۰ و حد بالای ۱ به‌طور تصادفی برای هر مؤلفه بردار انتخاب می‌شود. فلوچارت روش جستجوی هماهنگی ساده در شکل ۵ نشان داده شده است.

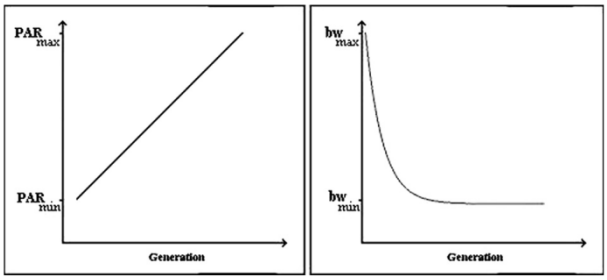
۰,۵۱	۰,۴۴	۰,۱۶	۰,۷۵	۰,۳۲
x_i^1	x_i^2	x_i^3	x_i^4	x_i^5

شکل ۴- i-امین بردار حل در حافظه الگوریتم

۳-۲-الگوریتم جستجوی هماهنگی بهبود داده شده[5](IHS)

در این الگوریتم روش جدیدی در ایجاد بردار حل جدید ارائه شده است که دقت و همگرایی آن نسبت به روش جستجوی هماهنگی ساده بهبود می‌یابد [۱۰]. نقش پارامتر *PAR* در افزایش تنوع در شروع جستجوی الگوریتم در فضای حل و نقش *BW* در جستجوی محلی الگوریتم به‌منظور افزایش نرخ همگرایی تأثیرگذار است.

1. Harmony memory consideration rate
2. Pitch adjustment rate
3. Band width
4. Number of Improvisation
5. Improved Harmony Search



شکل ۶: تغییرات پارامترهای PAR و BW در الگوریتم IHS [۱۰]

$BW_{max}, BW_{min}, PAR_{max}, PAR_{min}$، مقادیر ثابتی هستند که در ابتدای الگوریتم IHS تعریف می‌شوند. T شماره تکرار ایجاد بردار حل جدید الگوریتم است. NI شماره تعداد بهبودها برای رسیدن به توقف است.نشان داده شده است که در اکثر مسائل بررسی شده، IHS عملکرد بهتری نسبت به HS دارد [۱۰].

۳-۳- الگوریتم جستجوی هماهنگی بهترین کلی[1] (GBHS)

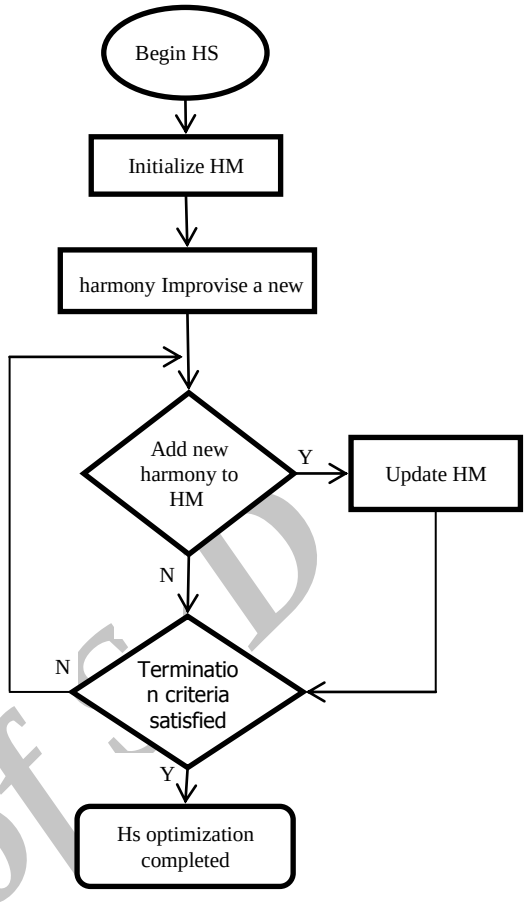
نتایج به دست آمده از این روش حل با نتایج حاصل از روش‌های IHS و HS مقایسه شد و در اکثر موارد نتایج بهتری به‌دست آمد [۱۱]. همچنین در ۶ تابع هدف که متغیرهای مؤلفه‌ها عدد صحیح بودند، نتایج این ۳ روش حل با یکدیگر مقایسه شد.

از مشکلات روش IHSاین است که باید مقادیر $BW_{max}, BW_{min}, PAR_{max}, PAR_{min}$، را برای مسأله به طور مناسب تعیین کنیم که تنظیم تعداد زیادی از پارامترها معمولا بر نتایج نهایی تأثیر منفی می‌گذارد. در این روش، قدم تنظیم گام (Pitch Adjustment) طوری طراحی شده که بردار جدید از بهترین بردار موجود در حافظه تقلید می‌کند. ایده این روش مشابه با روش PSO است که گروهی از ذرات در فضا حرکت می‌کنند و مؤلفه هر ذره از بهترین مؤلفه همان ذره و همچنین از مؤلفه بهترین ذره گروه تقلید می‌کند. پارامتر PAR مطابق با روش قبلی در هر تکرار به‌صورت پویا تغییر می‌کند و پارامتر BW از فرایند ایجاد حل جدید حذف شده و به جای آن فهم گروهی به کار می‌رود. شبه کد این روش مطابق شکل است.

همچنین، نتایج به‌دست آمده از روش پیشنهادی در ۹ تابع هدف بهینه‌سازی پیوسته بدون محدودیت با نتایج حاصل از IHSو HS مقایسه شد و در بیشتر موارد GBHS بهتر عمل نمود و تحلیل حساسیت برای تنظیم پارامترهای PAR، HMS و $HMCR$ انجام شد [۱۱].

۴-۳- الگوریتم جستجوی هماهنگی بهترین کلی جدید (NGHS)[2]

این الگوریتم را زو و همکاران [۱۳] معرفی کردند. یک ویژگی مهم PSO این است که از اجزای موفقش تقلید می‌کند. در این الگوریتم



شکل ۵: فلوچارت الگوریتم جستجوی هماهنگی ساده

از معایب روش HS استفاده از مقادیر ثابت PAR و BW است که تنظیم مناسب این پارامترها مشکل است. همچنین از معایب دیگر HS این است که تعداد تکرارهایی که الگوریتم نیاز دارد تا حل بهینه را پیدا کند مناسب نیست. اگر PAR مقدارش کوچک و BW مقدارش بزرگ باشد عملکرد الگوریتم ضعیف است و باید برای رفع آن تعداد بهبودها (NI) را افزایش دهیم تا به حل بهینه دست یابیم (شکل۶). هرچه مقدار پارامتر BW در تکرارهای ابتدایی بزرگتر باشد موجب می‌شود که الگوریتم، تنوع جستجو در کل فضای حل را افزایش دهد و در تکرارهای بعدی به منظور جستجوی محلی مقدار کمتر آن مناسب است. بنابراین مقادیر بزرگ PAR و مقادیر کوچک BW در تکرارهای پایانی منجر به رسیدن به فضای بهینه و همگرایی به بهینگی می‌گردد. IHS مشابه با HS است با کمی تفاوت که مطابق آن مقادیر پارامترهای PAR و BW به صورت پویا در هر تکرار جداگانه مطابق روابط زیر حاصل می‌گردند. شبه کد این روش حل مطابق شبه کد HS ساده است.

$$PAR(t) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{NI} \times t \qquad (۲)$$

$$BW(t) = BW_{\max} \times e^{\left(\frac{\ln(\frac{BW_{\min}}{BW_{\max}})}{NI} \times t\right)} \qquad (۳)$$

1. Global best harmony search
2. Novel Global Harmony search

یک تغییر در قدم سوم HS که ایجاد یک حل جدید است به وجود آمد که در آن حل جدید از بهترین حل موجود در حافظه HS تقلید می‌کند.

```
for (j=1 to n) do
    if (r1< HMCR) then
        Xnew(j) = Xa(j) where a ∈(1,2,…,HMS)
        if (r2< PAR) then
            Xnew(j) = XB(k), where k ∈(1,2,…,n)
        endif
    else
        Xnew(j) = LBj + r × (UBj -LBj), where r ∈(0,1)
    endif
endfor
```

شکل ۷: شبه کد روش GBHS [۱۱]

NGHS از هوش (فهم) گروهی الگوریتم PSO الهام گرفته است. دو عامل جدید *TRUST REGION* و *ADAPTIVE STEP* در NGHS طراحی شده است. بر پایه این دو عامل *position updating* جدید در NGHS برای حرکت دادن حل‌های نامناسب به سمت بهترین حل موجود حافظه در هر تکرار ایجاد می‌گردد. هدف از هر تکرار در الگوریتم NGHS، پیش بردن هماهنگی نامناسب در حافظه الگوریتم، به سمت بهترین هماهنگی کلی است. استفاده از قاعده به‌روزرسانی موقعیت بردارهای حل، نرخ همگرایی را تسریع می‌کند.

عیب الگوریتم HS یعنی نارس بودن همگرایی را با اضافه کردن احتمال جهش ژنتیکی P_m با مقدار احتمال کوچکاز بین خواهد برد. جهش ژنتیک با مقدار احتمال کوچک برای NGHS سودمند است چون هم تنوع حل‌های کاندید و هم ظرفیت جستجوی فضای حل را در NGHS افزایش می‌دهد.

NGHS و HS در موارد زیر با هم تفاوت دارند:

۱- در قدم اول: *HMCR* و *PAR* از NGHS حذف می‌گردند و احتمال جهش ژنتیک(*Pm*) به آن اضافه می‌گردد.

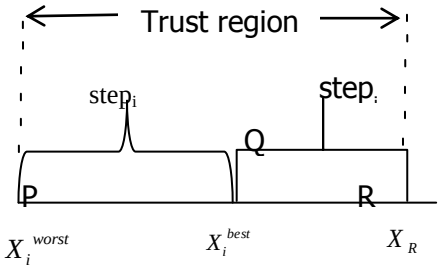
۲- در قدم سوم: NGHS قدم ایجاد حل جدید را از HS تقلید می‌کند و به صورت زیر شکل ۸ عمل می‌نماید.

```
for each i ∈ [1,N] do
    Stepi = |Xi^best − Xi^worst| % calculating the adaptive step
    Xi' = Xi^best ± r × stepi % position updating
    if rand ≤ Pm then
        Xi' = Xil + rand × (XiU -Xil) % genetic mutation
    end
end
```

شکل ۸: شبه کد ایجاد حل جدید با روش NGHS [۱۳]

در شکل ، $r1_i$، برای اجرای عملگر *updating position*، $r2_i$ برای تعیین اجرای پارامتر P_m و $r3_i$ برای mutation genetic استفاده می‌شوند. r_i به‌صورت تصادفی از یک توزیع احتمال یکنواخت در فاصله صفر و یک به‌دست می‌آید.

در شکل ۹، به عنوان قدم سازگار از i-امین متغیر تصمیم (i-امین مؤلفه هر هارمونی) تعریف می‌شود. محدوده بین نقاط Pو R به‌عنوان محدوده حقیقی برای i-امین متغیر تصمیم تعریف می‌شود. محدوده حقیقی یک منطقه نزدیک به بهترین حل هارمونی کلی است. تحلیل مناسب مطابق زیر است که در مرحله اولیه از بهینه‌سازی همه بردارهای حل در فضای حل پراکنده هستند. لذا، بیشتر قدم‌های سازگار و حدود حقیقی بزرگ هستند که این امر برای جستجوی سراسری (Global Search) در NGHS مناسب است. در حالی که در مراحل آخر بهینه‌سازی همه بردارهای حل که بهترین نیستند تمایل دارند به سمت بهترین بردار حل سراسری حرکت کنند، بنابراین اکثر بردارهای حل به هم نزدیک هستند. در این مورد اکثر قدم‌های سازگار و حدود حقیقی کوچکهستند که در NGHS برای جستجوی محلی مناسب است.



شکل ۹: منطقه حقیقی هر مؤلفه از یک بردار حل [۱۳]

طراحی درست برای قدم سازگار می‌تواند تضمین کند که الگوریتم پیشنهادی توانایی جستجوی سراسری قوی در قدم‌های اولیه از بهینه‌سازی و توانایی جستجوی محلی قوی در مراحل پایانی بهینه‌سازی را دارد. تنظیم کردن $step_i$ تعادل بین جستجوی سراسری و محلی را حفظ می‌کند.عملگر P_mبا یک احتمال کوچک برای بهترین حل از حافظه هارمونی بعد از Position Updating به کار می‌رود تا ظرفیت دور شدن از بهینگی محلی در الگوریتم افزایش یابد.

۳-در قدم چهارم $X^{'}$ جایگزین X^{WORST} در حافظه می‌گردد حتی اگر $X^{'}$ بدتر از X^{WORST} باشد (یعنی برخلاف روش HS، مقدار برازش تابع هدف $X^{'}$ کمتر از X^{WORST} باشد بازهم جایگزین آن در حافظه الگوریتم می‌شود).

۳-۵- الگوریتم جستجوی هماهنگی کلی خودانطباقی (SGHS)[1]

این روش در پژوهش مرجع [۱۲] ارائه شده است. SGHS از روش ارائه شده در پژوهش [۱۱] الهام گرفته شده است. در این الگوریتم یک روش جدید برای ایجاد حل جدید و روش‌های جدیدی برای

1. Self-Adaptive Global Harmony Search

تنظیم پویای پارامترهای الگوریتم معرفی شد. مشابه با روش GBHS از خصوصیات بهترین حل در حافظه پیروی می‌کند و مؤلفه تکرار فعلی از بهترین بردار حل حافظه به صورت تصادفی انتخاب می‌شود که ممکن است حل جدید ساختار نامناسبی پیدا کند و به خوبی بردار X_B نباشد و بنابراین در این روش مطابق (۴ از مؤلفه مشابه با مؤلفه بهترین حل در ایجاد حل جدید استفاده می‌کنیم.

$$X_{new}(j) = X_B(j) \qquad (۴)$$

تفاوت دیگر این روش در تولید حل جدید این است که برای نیفتادن الگوریتم در دام بهینگی محلی، در قاعده اول (در نظرگیری حافظه) از (۵ استفاده می‌شود.

$$X_{new}(j) = X_a(j) \pm r \times BW \qquad (۵)$$

SGHS به تنظیم دقیق پارامترهایش در مسائل مختلف نیاز ندارد چون دارای مکانیزم یادگیری در تعیین پارامترهای *HMCR*(*PAR*) و تنظیم پارامتر *BW* به صورت پویای کاهشی با شماره نسل است. رویه این روش حل به صورت زیر است:

قدم اول: پارامترهای *HMS*، *LP*و *NI* تعیین می‌شود.

قدم دوم: مقادیر اولیه $HMCR_m$، PAR_m، BW_{MIN} و BW_{MAX} تخصیص می‌یابند.

قدم سوم: حافظه هارمونی مقداردهی اولیه شده و شماره نسل برابر lp=1 می‌گردد.

قدم چهارم: مطابق با $HMCR_m$و PAR_m مقادیر *HMCR*و *PAR* تعیین می‌شود.

قدم پنجم: مطابق شبه کد شکل ۱۰ بردار حل جدید ایجاد می‌شود.

قدم ششم: اگر$F(X_{new}) \leq F(X_{worst})$ باشد حافظه به روز و مقادیر*HMCR* و *PAR*ثبت می‌شود.

```
for (j=1 to n) do
    if (r1< HMCR) then
        Xnew(j) = Xa(j) ± r × BW where r ∈ (0,1)
        if (r2< PAR) then
            Xnew(j) = XB(j) where r1,r2 ∈ (0,1)
        endif
    else
        Xnew(j) = LBj + r × (UBj -LBj), where r ∈ (0,1)
    endif
    endfor
```

شکل ۱۰- شبه کد ایجاد حل جدید در الگوریتم SGHS [۱۲]

قدم هفتم: اگر $lp= LP$باشد مطابق با $HMCR_m$ و PAR_mثبت شده، مقادیر *HMCR* و *PAR* تعیین شده وlp=1می‌شود، در غیر این صورت $lp= lp+1$ خواهد بود.

قدم هشت: اگر تعداد تکرارها برابر *NI*شده باشد بهترین حل حافظه ثبت می‌شود، در غیر این صورت به قدم چهارم می‌رود.

- **روش کلی الگوریتم SGHS**

فرض شود که *HMCR* (*PAR*) دارای توزیع نرمال در بازه[0.9,1] ([0,1]) با متوسط $HMCR_m$ (PAR_m) و انحراف معیار ۰,۰۱ (۰,۰۵) باشد. ابتدا $HMCR_m$ (PAR_m) را ۰,۹۸ (۰,۹) تعیین و سپس مطابق با توزیع نرمال پارامترهای مورد نظر مقداردهی می‌شوند. بعد از طی تکرارهایی مقدار *HMCR* (*PAR*) حل‌های بهتر که جایگزین حل‌های بدتر در حافظه می‌شوند ثبت می‌شوند. بعد از یک تعداد مشخص از تکرارهای دوره یادگیری (Learning Period)، از میانگین *HMCR*(*PAR*) ثبت شده در دوره، مقادیر $HMCR_m$ (PAR_m) را محاسبه می‌کنیم. با توجه به این مقادیر متوسط و همچنین انحراف معیار با توزیع نرمال مقادیر *HMCR (PAR)* جدید تولید می‌شود و در تکرارهای بعدی استفاده می‌شوند. این روش تکرار می‌گردد و در طی این فرایند مقدار مناسب *HMCR* (*PAR*) یادگیری می‌شود تا با مسأله مورد مطالعه و فرایند جستجو تطبیق یابد.

پارامتر *BW* در این الگوریتم به صورت پویا تنظیم می‌شود. برای تعادل بین جستجوی کلی و جستجوی محلی در فرایند جستجو در تکرارهای مختلف از (۶زیر استفاده شد:

$$BW(t) = \begin{cases} BW_{\max} - \dfrac{BW_{\max} - BW_{\min}}{NI} \times 2t \\ BW_{\min} \end{cases} \qquad (۶)$$

۳-۶- الگوریتم جستجوی هماهنگی با زیر نسل پویا (DSHS)[1]

این روش در پژوهش [۱۲] ارائه شد. حافظه الگوریتم به صورت پویا تغییرمی‌کند. به عبارت دیگر کل حافظه الگوریتم به زیرمجموعه‌هایی با اندازه کوچکتر از *HMS* تقسیم می‌شود. هر زیر مجموعه حافظه، سیر تکاملی به‌منظور پیدا کردن بهترین حل را اجرا می‌کند و به صورت دوره‌ای اطلاعاتش را با دیگر زیرمجموعه‌ها با استفاده از دوباره گروه‌سازی تبادل می‌کند تا در بهینگی محلی نیافتد. همچنین، در این الگوریتم در قسمت ایجاد حل جدید از فرایند حل جدید با استفاده از اطلاعات بهترین حل محلی در هر زیرجمعیت استفاده می‌شود. هر چه اندازه تعداد بردارها در هر زیرمجموعه کمتر باشد سرعت همگرایی و تنوع متعادل‌تر است. هر *R* تکرار، کل بردارهای حل در حافظه، مجددا به صورت تصادفی به گروه‌های زیر مجموعه طبقه‌بندی می‌شوند و در ادامه همان روند نسل قبلی تا *R* تکرار ادامه می‌یابد. اطلاعات به دست آمده هر زیر مجموعه در فرایند جستجوی قبلی در میان حل‌ها مبادله می‌شود که تنوع هر زیر مجموعه افزایش می‌یابد.

در قسمت تولید حل جدید آن بردار حل‌هایی که عملکرد بهتری

1. Harmony search algorithm with Dynamic sub-population

دارند با احتمال بالاتری برای بردار جدید انتخاب می‌شوند به همین دلیل در مراحل ابتدایی به همگرایی زودرس (بهینگی محلی) می‌رسد. در این پژوهشاز قاعده تنظیم گام جدید، برای افزایش تنوع و کاهش نرخ همگرایی به صورت (۷ ارائه می‌شود:

$$X_{new}(j) = X_a(j) \pm r \times BW \qquad (۷)$$

پارامترa با روش چرخ رولت هر بردار حل که در حافظه قرار دارد انتخاب می‌شود که اگر هر حل در حافظه از نظر مقدار تابع هدف بهتر باشد احتمال بیشتری برای انتخاب شدن دارد تا در فرایند ایجاد حل جدید نقش بیشتری داشته باشد.

- **گام‌های الگوریتم DSHS**

۱) تعیین پارامترهای $HMCR$، PAR، HMS، BW، NI، S، R، K=0

۲) مقداردهی اولیه حافظه هارمونی

۳) تقسیم حافظه به تعدادی زیر مجموعه که اندازه هر زیرمجموعه S باشد.

۴) مراحل زیر تکرار شود تا وقتی که K برابر با NI شود.

۴-۱) برای هر زیرمجموعه در حافظه قدم‌های زیر اجرا می‌شود:

- یک حل جدید ایجاد می‌شود.
- اگر مقدار برازش حل جدید از بدترین حل در هر زیرمجموعه بهتر بود حل جدید جایگزین آن می‌شود.

۴-۲) اگر $K \bmod (K,R)=0$ کل بردارهای حل در حافظه را به زیرمجموعه‌ها گروه‌بندی شود.

۴-۳) $K=K+1$

۵) بهترین بردار حل موجود در حافظه را خارج کنیم.

۳-۷- الگوریتم جستجوی هماهنگی با قابلیت اطمینان بالا [1]

روش فوق در پژوهش [۱۴] ارائه شده است. این روش مشابه با روش IHS است ولی روش محاسبه PAR مطابق (۸ است. مطابق با این روش هرچه مقدار PAR در ابتدای تکرارهای الگوریتم (نسل‌های اولیه) بیشتر باشد، تنوع جستجوی کل فضای جواب مسأله بیشتر و هرچه به نسل‌های انتهایی می‌رویم مقدار PARکمتر، تنوع کمتر و جستجوی محلی بیشتر می‌شود:

$$PAR(gn) = PAR_{\max} - \frac{PAR_{\max} - PAR_{\min}}{NI} \times gn \qquad (۸)$$

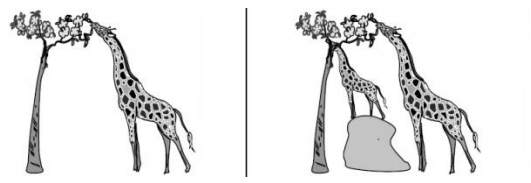
۴- الگوریتم جستجوی هماهنگی پیشنهادی [2] (TNHS)

این الگوریتم بعد از تحقیقات گسترده بر روی روش‌های مرور شده در ادبیات موضوع روش جستجوی هماهنگی ارائه شد. در این روش، نوآوری‌های جدید در قسمت تنظیم پارامترها و تغییراتی در روند مقدار پارامترهای $HMCR$ و PAR در طول تکرارهای الگوریتم در مقایسه با روش GBHS ارائه شده است. از آنجایی که الگوریتم GBHS در مقایسات نتایج حاصل از چندین آزمایش نسبت به روش‌های HS و IHS بهتر عمل نمود، در نتیجه این الگوریتم به‌عنوان الگوریتم پایه در نظر گرفته می‌شود و با تغییر در پارامترها و عملگرهای آن سعی در رسیدن به روشی کاراتر از آن به‌منظور حل مسائل بهینه‌سازی داریم.

الگوریتم‌های فراابتکاری همانند ابزارهای قدرتمند به منظور جستجوی بهترین حل در مسائل بهینه‌سازی در حوزه‌های مختلف مورد استفاده قرار گرفته است. با وجود کارا بودن الگوریتم‌های فراابتکاری در حل مسائل تصمیم‌گیری و بهینه‌سازی، این الگوریتم‌ها نیاز به دقت و ظرافت فراوانی در هنگام اجرای آزمایش‌ها دارند. بهکارگیری نامناسب و تنظیم نامناسب پارامترهای آن سبب می‌شود که کارایی و اثربخشی این روش‌ها کاهش یابد. منظور از تنظیم پارامتر آن است که، انتخاب بهترین حالت یا مقدار برای پارامترها به نحوی که عملکرد الگوریتم را در سطح بهینه آن (بهترین عملکرد ممکن الگوریتم) قرار دهد. این پارامترها ممکن است تأثیر زیادی بر کارایی و اثربخشی الگوریتم داشته باشد.

در جستجوی حل‌های بهینه روشی کاراتر است که در ابتدای جستجو، تنوع کل فضای حل را در نظر بگیرد (Exploration) یعنی کل فضای حل را شناسایی کند و بیشتر به سمت فضاهایی که با احتمال بیشتری بهینگی در آن قرار دارد حرکت کند.سپس با دو نوع بهینگی ممکن است مواجه شود: بهینگی محلی و بهینگی سراسری.

در صورتی که الگوریتم در دام بهینگی محلی قرار بگیرد می‌تواند با استفاده از تکنیک‌های جستجوی محلی از آن منطقه به صورتی جدا شود. که مطابق با شکل ۱۱ حل‌ها با سرعت بالاتری به مقدارهدف می‌رسند.مطابق با توضیحات ارائه شده در این بخش، الگوریتم HS این امکان را دارد که در زمان مناسبی محدوده‌های عملکردی بالا در فضای حل راشناسایی کند اما در اجرای جستجوی محلی در مسائل بهینه‌سازی ترکیباتی، کارا نیست که در این مواقع در بهینگی محلی قرار می‌گیرد. یکی از تکنیک‌های مورد استفاده جهت غلبه بر این مشکل، روش فاز restart است. با این روش یک شوک به مکانی که اکنون الگوریتم در فضای حل مسأله قرار دارد وارد می‌کند و سبب افزایش پراکندگی می‌شود.



شکل ۱۱: نقش جستجوی محلی در پیدا کردن حل بهینه

در این روش هنگامی که بعد از طی تکرارهایی بهترین مقدار بهبود نیابد به صورت زیر عمل می‌کند:

[1] Highly Reliable Harmony Search Algorithm
[2] Hybrid Taguchi and Novel global best harmony search

گام۱. حافظه هارمونی بر اساس بهترین مقدار تابع هدف به صورت افزایشی مرتب می شود.

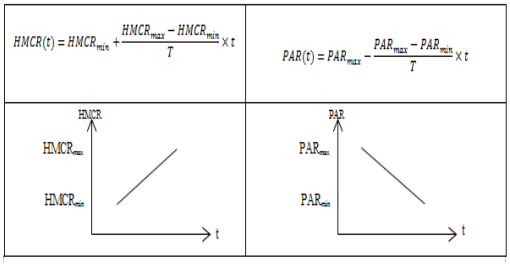
گام۲. به اندازه‌ی درصد درنظرگیری فاز restart(*Restart phase consideration rate%*) حل‌های اول لیست حافظه مرتب شده نگهدار می‌دارد، سپس برای *Restart phase consideration rate*-۱حافظه باقیمانده، نیمی را با استفاده از عملگر جهش تک نقطه‌ای روی *Regenerate Restart phase rate* حافظه انتخابی و نیمی دیگر را بر اساس محدوده مؤلفه‌های هر حل به طور تصادفی ایجاد می‌کند.

۴-۱- تنظیم پارامترها در الگوریتم جستجوی هماهنگی پیشنهادی

در الگوریتم جستجوی هماهنگی پیشنهادی تعدادی پارامترشامل تعداد بردارهای حافظه الگوریتم (*hms*)، حداقل نرخ تنظیم گام (PAR_{min})، حداکثر نرخ تنظیم گام (PAR_{max})، حداقل نرخ در نظرگرفتن حافظه ($hmcr_{min}$) و حداکثر نرخ در نظرگرفتن حافظه ($hmcr_{max}$) است که باید با مقادیر مناسب تنظیم شوند. منظور از تنظیم پارامتر، انتخاب بهترین مقدار برای پارامترها به نحوی است که عملکرد الگوریتم در سطح بهینه قرار گیرد. لذا، پارامترهای هر الگوریتم تأثیر فراوانی بر کارایی و اثربخشی الگوریتم دارند. تنظیم نامناسب پارامترها ممکن است موجب به‌دست آمدن نتایج نامناسبی در مسأله مورد مطالعه گردد. پارامترهای مناسب الگوریتم در یک مسأله، دلیل بر مناسب بودن آن‌ها در حل مسأله دیگری نخواهد بود و باید در هر مسأله به‌طور جداگانه پارامترها تنظیم شوند. در الگوریتم پیشنهادی، ۳ نوع تنظیم پارامتر وجود دارد که شامل تنظیم پویا، تنظیم انطباقی و تنظیم ترتیبی به روش تاگوچی است.

- **تنظیم پویا**

پارامترهای نرخ تنظیم گام (*PAR*) و نرخ در نظر گرفتن حافظه هماهنگی (*HMCR*) به صورت پویا در طول فرایند جستجو که با فرمول ریاضی در شکل ۱۲ ارائه شده، به صورت خطی تغییر می‌کنند و در این پارامترها در طول فرایند جستجو، مقادیر مختلفی از پارامترها مناسب بوده و یک مقدار ثابت در طول کل فرایند جستجو بهترین مقدار نیست.



شکل ۱۲: رابطه تغییرات پارامتر PAR و HMCR با افزایش نسل‌ها
T: زمان توقف الگوریتم، t: زمان سپری شده اجرای الگوریتم

- **تنظیم انطباقی**

مقدار تابع هدف بردارهای حل در حافظه، نسل به نسل با افزایش تکرارهای الگوریتم به دلیل داشتن حافظه در ساختار الگوریتم و نقش آن در جهت بهبود روند جستجو، تا رسیدن به شرط توقف بدتر نمی‌شوند.

- **تنظیم ترتیبی با روش تاگوچی**

ترکیب مناسب پارامترها در کیفیت جواب نهایی الگوریتم تأثیر فراوانی دارد از این رو به‌منظور مشخص کردن پارامترهای بهتر، بهترین ترکیب پنج پارامتر محاسبه شود. با توجه به سطوح بهینه هر پارامتر که ۳ سطح است در مجموع 3^5 ترکیب پارامتر جهت هر اجرای مسأله وجود دارد که با توجه به‌تعداد نمونه مسائل(۹) و تکرار ۵ مرتبه از هر آزمایش که در بخش ۵ توضیح داده می‌شود، کل آزمایش‌ها $3^5*5*9=10935$ می‌شود که معین کردن بهترین ترکیب با این تعداد آزمایش، مستلزم صرف زمان بسیار زیاد و نامعقول است. لذا، از تکنیک طراحی آزمایش تاگوچی استفاده می‌شود. شکل ۱۳ فلوچارت الگوریتم جستجوی هماهنگی پیشنهادی را نشان می‌دهد.

روش پیشنهادی در این پژوهش در جدول ۱ با روش‌های HS، IHS و GBHS مقایسه و نوآوری‌ها و نوع تغییرات پارامترها بررسی شده است.

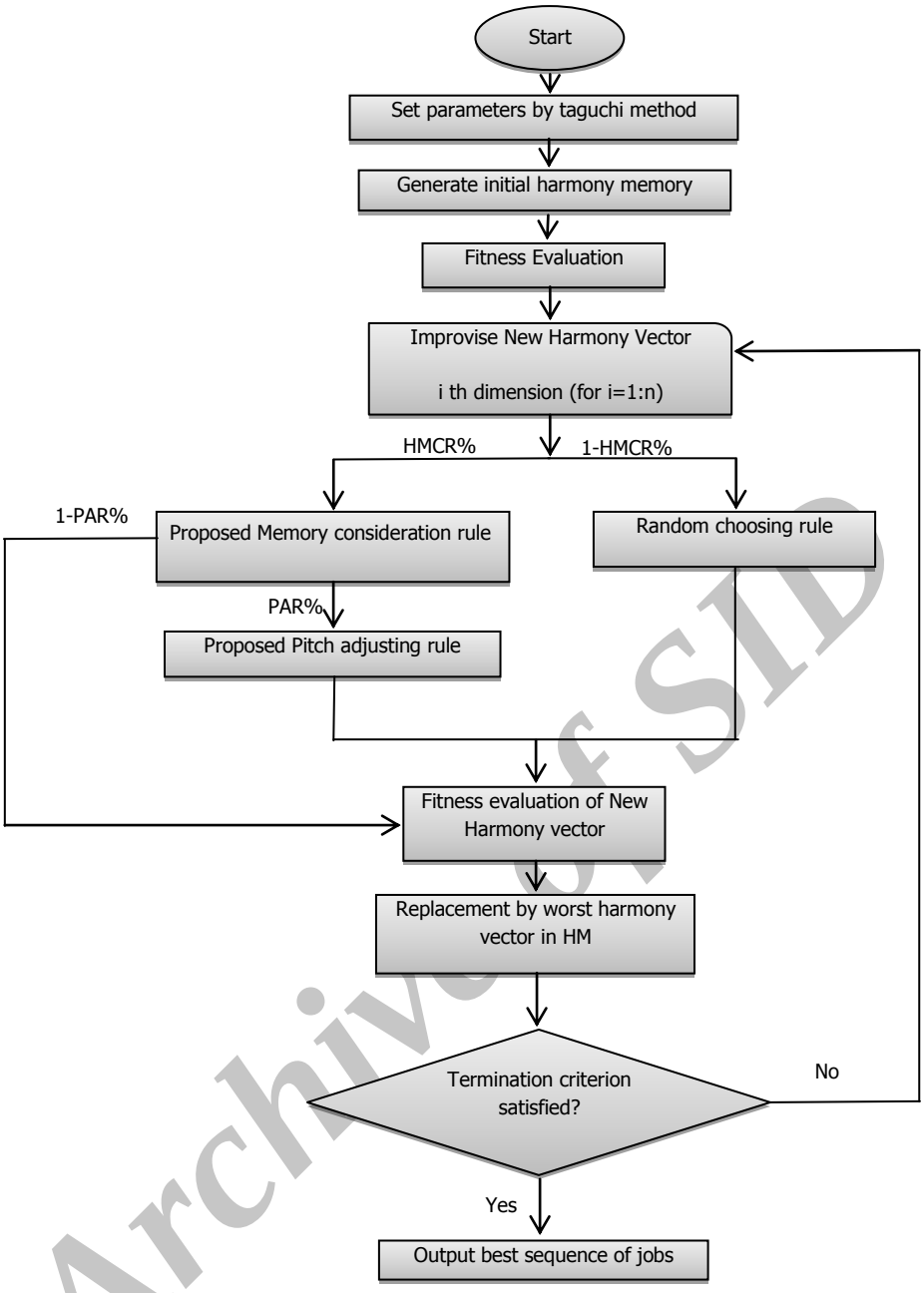
۵- نتایج محاسباتی

۵-۱- نمونه مسائل

برای آزمایش عملکرد الگوریتم ارائه شده، ارزیابی گسترده ریاضیاتی در مقایسه با الگوریتم‌های IHS, HS و GBHS بر روی مجموعه مسائل بهینه سازی زیر اجرا شده است. جدول ۲ مجموعه توابع پیوسته بهینه‌سازی را نشان می‌دهد. این توابع از مرجع [۱۵] گرفته شده است.

۵-۲- تنظیم فاکتور و سطوح هر فاکتور الگوریتم‌های حل مدل

به منظور حل ۹ مسأله، چهار الگوریتم فراابتکاری شامل جستجوی هماهنگی ساده، جستجوی هماهنگی بهبود یافته، جستجوی هماهنگی بهترین کلی، جستجوی هماهنگی پیشنهادی(TNHS) با هم مقایسه می‌شوند. برای هر کدام از این الگوریتم‌ها، با استفاده از آزمایش‌های مقدماتی پارامترهای مناسبی انتخاب شده‌اند، که در جدول‌های ۳ تا ۶ نشان داده شده است. سپس با استفاده از روش تاگوچی بهترین پارامتر و بهترین ترکیب از پارامترها و عملگرها انتخاب می‌شوند.



شکل ۱۳: فلوچارت الگوریتم جستجوی هماهنگی پیشنهادی (TNHS)

جدول ۱:مقایسه الگوریتم پیشنهادی با سایر روش‌های جستجوی هماهنگی

	Parameters				Innovation
	HMS	HMCR	PAR	BW	
HS	Constant	Constant	Constant	Constant	Imitating the improvisation process of musicians [۱]
IHS	Constant	Constant	Dynamic-Ascending	Exponential	IHS employs a novel method for generating new solution vectors [10]
GBHS	Constant	Constant	Dynamic-Ascending	Nothing	concepts from intelligence swarm are borrowed [11]
TNHS (present research)	Constant	Dynamic-Ascending	Dynamic-descending	Nothing	1-hybrid restart phase as a local search and GBHS
					2-all parameters are tuned with taguchi method

جدول ۲:توابع هدف مورد آزمایش برای مقایسه الگوریتم‌ها

ردیف	تابع هدف	فرمول ریاضی	حدود x	مقدار بهینه f(x)
۱	Sphere	$\sum_{i=1}^{p} x_i^2$	[-5.12,5.12]	0
۲	Schwefel's 2.22	$\sum_{i=1}^{n} \lvert X(i)\rvert + \prod_{i=1}^{n} \lvert X(i)\rvert$	[-10,10]	0
۳	Rosenberg	$\sum_{i=1}^{n-1} (100(X(i+1) - X^2(i))^2 + (X(i)-1)^2)$	[-30,30]	0
۴	Step	$\sum_{i=1}^{n} \left(\lfloor X(i) + 0.5 \rfloor\right)^2$	[-100,100]	0
۵	Rotated hyper-ellipsoid	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} X(j) \right)^2$	[-100,100]	0
۶	Schwefel's 2.26	$418.9829\,n - \sum_{i=1}^{n} \left(X(i) \sin\left(\sqrt{\lvert X(i)\rvert}\right)\right)$	[-500,500]	0
۷	Rastrigin	$\sum_{i=1}^{n} X^2(i) - 10\cos(2\pi X(i)) + 10)$	[-5.12,5.12]	0
۸	Ackley's	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} X^2(i)}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi X(i))\right) + 20 + e$	[-32,32]	0
۹	Griewank	$\frac{1}{4000}\sum_{i=1}^{n} X^2(i) - \prod_{i=1}^{n} \cos\left(\frac{X(i)}{\sqrt{i}}\right) + 1$	[-600,600]	0

جدول ۳: فاکتورها و سطوح کاندید در الگوریتم جستجوی هماهنگی (HS)

پارامتر HS	نماد HS	سطوح HS
اندازه حافظه الگوریتم	HMS	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15
نرخ در نظرگیری حافظه الگوریتم	HMCR	HMCR(1) : 50% HMCR(2) : 80% HMCR(3) : 99%
نرخ تنظیم گام	PAR	PAR(1) : 10% PAR(2) :50% PAR(3) : 90%
پهنای باند	BW	BW(1) : 0.2 BW(2) : 0.5 BW(3) : 0.99

جدول ۴: فاکتورها و سطوح کاندید در الگوریتم جستجوی هماهنگی بهترین کلی (GBHS)

پارامتر GBHS	نماد GBHS	سطوح GBHS
اندازه حافظه الگوریتم	HMS	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15
نرخ در نظرگیری حافظه الگوریتم	HMCR	HMCR(1) : 30% HMCR(2) : 60% HMCR(3) : 98%
حداقل نرخ تنظیم گام	PAR_{min}	PAR_{min}(1) : 0 PAR_{min}(2) : 20% PAR_{min}(3) : 49%
حداکثر نرخ تنظیم گام	PAR_{max}	PAR_{max}(1) :50% PAR_{max} (2) :70% PAR_{max} (3) :90%

جدول ۵: فاکتورها و سطوح کاندید در الگوریتم جستجوی هماهنگی بهبودیافته (IHS)

پارامتر HIS	نماد IHS	سطوح IHS
اندازه حافظه الگوریتم	HMS	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15
نرخ در نظرگیری حافظه الگوریتم	HMCR	HMCR(1) : 50% HMCR(2) : 80% HMCR(3) : 99%
حداقل نرخ تنظیم گام	PAR_{min}	PAR_{min}(1) :20% PAR_{min}(2) :40%
حداکثر نرخ تنظیم گام	PAR_{max}	PAR_{max}(1) :50% PAR_{max} (2) :70% PAR_{max} (3) :90%
حداقل پهنای باند	BW_{min}	BW_{min}(1) : 0.2 BW_{min}(2) : 0.4
حداکثر پهنای باند	BW_{max}	BW_{max}(1) : 0.5 BW_{max}(2) : 0.99

جدول۶:فاکتورها وسطوح کاندید در الگوریتم جستجوی هماهنگی پیشنهادی (TNHS)

پارامتر TNHS	نماد TNHS	سطوح TNHS
اندازه حافظه الگوریتم	HMS	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15
حداقل نرخ در نظرگیری حافظه الگوریتم	$HMCR_{min}$	$HMCR_{min}$(1) : 20% $HMCR_{min}$(2) : 50%
حداکثر نرخ در نظرگیری حافظه الگوریتم	$HMCR_{max}$	$HMCR_{max}$(1) : 80% $HMCR_{max}$(2) : 99%
حداقل نرخ تنظیم گام	PAR_{min}	PAR_{min}(1) : 20% PAR_{min}(2) : 40%
حداکثر نرخ تنظیم گام	PAR_{max}	PAR_{max}(1) :50% PAR_{max} (2) :90%

۵-۳- انتخاب بهترین فاکتورها

در هر اجرای آزمایش مقدار تابع هدف به دست آمده باید مطابق روش تاگوچی به نسبت سیگنال به نویز که در حکم متغیر پاسخ است تبدیل شود و مطابق تغییرات آن تحلیل صورت گیرد. از آنجا که هدف هر اجرا حداقل کردن تابع هدف است، لذا نوع سیگنال به نویز زیر انتخاب می‌شود.

$$S/N_s = -10\ \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \qquad (۹)$$

در روش تاگوچی نسبت S/N در حکم متغیر نسبت است که تابع هدف در هر اجرا به این نسبت تبدیل می‌شود تا بر طبق آن تصمیم‌گیری شود. در این پژوهش با توجه به نسبت S/N_s انتخاب شده مناسب ماهیت مسائل این پژوهش، کمترین نسبت S/N_sبرای هر فاکتوردر هر الگوریتم، به عنوان فاکتور بهینه انتخاب می‌شود.

۵-۴- انتخاب فاکتورهای بهینه الگوریتم‌های حل

بعد از اجرای آزمایش‌ها، نتایج حاصل از کل اجراهای الگوریتم-هابه روش طراحی آزمایش‌های تاگوچی جهت تنظیم پارامترها، در جدول ۷ نشان داده شده‌اند.

۵-۵- اجرای آزمایش‌ها

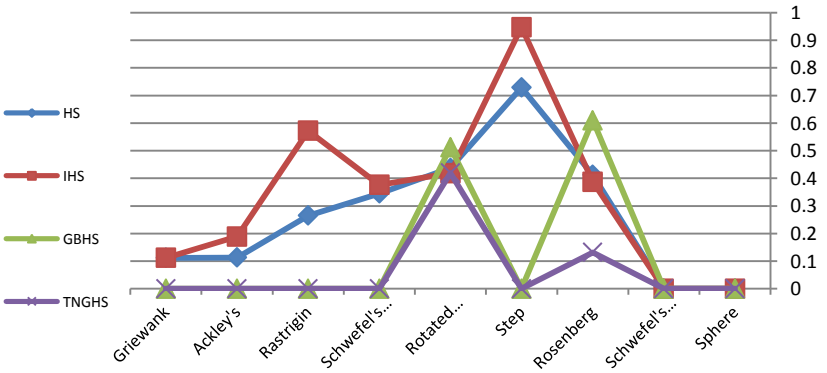
برای اجرای آزمایش‌ها، همه‌ی الگوریتم‌های به کار رفته در این پژوهش با نرم‌افزار متلب روی کامپیوتر شخصی با ریزپردازنده 2.27 core iGHz 5 و 4.00 GBحافظه، برنامه‌نویسی و سپس اجرا شده است. نتایج حاصل از حل در جدول ۸ و **شکل** ۱۴ آمده است.

جدول ۷:پارامترهای بهینه الگوریتم‌های حل با روش تاگوچی

	اندازه حافظه الگوریتم	نرخ در نظرگیری حافظه الگوریتم		نرخ تنظیم گام		پهنای باند	
		حداکثر	حداقل	حداکثر	حداقل	حداکثر	حداقل
HS	۵	۰,۸		۰,۱		۰,۲	
IHS	۱۰	۰,۸		۰,۷	۰,۲	۰,۹۹	۰,۴
GBHS	۱۵	۰,۶		۰,۹	۰	-	
TNHS	۱۵	۰,۸	۰,۲	۰,۵	۰,۲	-	

جدول ۸: نتایج حاصل از حل ۵ مرتبه از هر مسأله در هر الگوریتم (میانگین)

تابع هدف	مقدار بهینه سراسری	HS	IHS	GBHS	TNGHS
Sphere	۰	۰,۰۰۰۱۴۸	۰,۰۰۰۳۳۱	۰,۰۰۰۱۰۱	**۰,۰۰۰۰۱۱**
Schwefel's 2.22	۰	۰,۰۹۲۷۸۱	۰,۱۷۳۲۶۶	۰,۰۶۳۹۲۱	**۰,۰۰۲۱۳۲**
Rosenberg	۰	۴۱۲,۴۷۷۱	۳۸۷,۶۴۹۳	**۶۱,۰۲۹۴۸**	۱۳۱,۹۰۲۶
Step	۰	۷,۲۹۱۸	۹,۴۷۸۱	۰,۰۰۰۴	**۰,۰۰۰۰**
Rotated hyper-ellipsoid	۰	۴۳۷۱,۵۸۱۹	**۴۱۸۸,۷۳۱۵**	۵۱۱۸,۶۳۷۲	۴۱۹۴,۵۰۱۴
Schwefel's 2.26	۰	۳۴,۵۲۸۵	۳۷,۶۳۸۱	۰,۰۹۲۱	**۰,۰۰۲۸۱**
Rastrigin	۰	۲,۶۵۱۰	۵,۷۲۱۹	**۰,۰۰۹۵**	۰,۰۲۱۸
Ackley's	۰	۱,۱۳۰۰	۱,۸۹۳۳	**۰,۰۲۰۹**	۰,۷۸۴۴
Griewank	۰	۱,۱۱۹۲	۱,۱۲۰۹	۰,۱۰۲۴	**۰,۰۵۲۷**



شکل ۱۴: نمودار مقایسه نتایج حاصل از حل هر مسأله در هر الگوریتم

برای رسم نمودار شکل ۱۴ با توجه به اینکه در جدول ۸ تفاوت بزرگ مقداری بین نتایج حاصل از توابع هدف مختلف وجود دارد، نتایج تابع هدف Rotated hyper بر عدد 10^4 و تابع هدف Rosenberg بر عدد 10^3 تقسیم می‌شود. با توجه به نتایج به دست آمده در ۵۵ درصد موارد از مقایسات، روش جستجوی هماهنگی ارائه شده در این پژوهش نسبت به ۳ روش دیگر از عملکرد بهتری برخوردار است.

۶- حل مسأله زمان‌بندی با روش جستجوی هماهنگی پیشنهادی

یکی از مهم‌ترین تصمیم‌ها در طراحی روش فراابتکاری این است که حل‌های مسأله به چه صورت نمایش داده شوند و چگونه به حل‌های واقعی ساختار مسأله تبدیل کنیم. نمایش جواب باید به شکلی باشد که موجب کاهش هزینه‌ها و زمان استفاده از الگوریتم شود. در این مسأله هر حل به‌صورت یک بردار nمؤلفه‌ای است. n تعداد کارهای حل در مسأله است. شیوه نمایش جواب در الگوریتم جستجوی هماهنگی توسعه داده شده بدین‌صورت است که هر مؤلفه بر اساس مقادیر تصادفی در بازه [۰,۱] به‌دست می‌آیدکه با قاعده موقعیت بزرگ‌ترین مقدار[1](LPV) به جایگشت از کارها تبدیل می‌شود که در مثال شکل ۱۵ نشان داده شده است.

۰,۶۴	۰,۸۱	۰,۹۱	۰,۴۶	۰,۵۲	۰,۳۱	۰,۲۵

یک بردار هارمونی تصادفی از اعداد در الگوریتم جستجوی هماهنگی

۰,۸۱	۰,۶۴	۰,۵۲	۰,۴۶	۰,۳۱	۰,۲۵	۰,۱۹

مرتب کردن رشته اعداد بردار بالا به صورت نزولی

۲	۱	۵	۴	۶	۷	۳

شکل ۱۵: به‌دست آوردن یک جایگشت ازاعداد مرتب شده تصادفی با قاعده LPV

۶-۱- ایجاد مسائل نمونه

در این بخش برای ارزیابی روش فراابتکاری پیشنهادی، مسائل نمونه با استفاده از مسائل موجود در ادبیات ایجاد می‌شود.

برای ایجاد مسائل نمونه از روشی مشابه کار ربادی و همکاران در سال ۲۰۰۶استفاده شده است. داده‌های مورد نیاز برای ارزیابی شامل تعدادی ماشین (n)، تعدادیکار (m) و زمان پردازش (p_j)است. در اینجا ۳ سطح برای تعداد کار{۶۰،۴۰،۲۰} و برای تعداد ماشین ۳ سطح {۶،۴،۲} در نظر گرفته شده است. زمان پردازش به‌طور تصادفی از بازه [۵۰,۱۰۰]ایجاد شده است. برای سرعت ماشین نیز فرض شده که سرعت اولین ماشین v_1=۱ و سرعت ماشین‌ها ازv_2تا v_mبه صورت زیر است:

$$v_m=v_1 + (m-1)\times 0.2 \qquad (۱۰)$$

مسائل به صورت ترکیبات mو nیعنی n×mنشان داده شده است.به عنوان مثال مسأله با ۴ ماشین و۲۰ کار به صورت ۴×۲۰ نشانداده می‌شود.

۶-۲- ارزیابی نتایج

دراین بخش هدف بررسی عملکرد روش فراابتکاری پیشنهادی است. به این منظور، آزمایش‌ها روی مسائل نمونه ایجاد شده، اجرا شده است.به منظور اطمینان بیشتر نتایج حاصل از اجرای الگوریتم جستجوی هماهنگی، از هر مسأله نمونه، ۳ مسأله تصادفی ایجاد و ۳ بار اجرا شده است که بهترین جواب هر مسأله از بین ۳ تکرار در جدول ۹ آورده شده است.در این جدول،i1 تا i3 شماره مسأله تصادفی ایجاد شده از هر مسأله نمونه است.

جدول ۹: نتایج حاصل از آزمایش مسائل نمونه

n×m	TNHS			HS		
	i1	i2	i3	i1	i2	i3
20×2	1995	1982	1986	1998	1987	1988
20×4	1348	1368	1362	1427	1421	1419
20×6	722	781	739	740	771	782
40×2	3833	3978	3913	3862	3991	3921
40×4	2581	2352	2178	2041	2218	2148
40×6	1349	1315	1368	1388	1371	1309
60×2	5870	5876	5764	5892	5882	5921
60×4	3143	3092	3122	3156	3052	3113
60×6	2067	2203	2018	2215	2242	2148

با توجه به نتایج به دست آمده از در اکثر موارد از مقایسات، روش جستجوی هماهنگی ارائه شده در این پژوهش نسبت به روش جستجوی هماهنگی ساده از عملکرد بهتری برخوردار است.

۷- نتیجه‌گیری

در این مقاله یک روش جدید در ساختار حل الگوریتم جستجوی هماهنگی ارائه شده است که از آن برای حل مسائل بهینه‌سازی پیوسته ریاضی استفاده می‌شود. با توجه به ویژگی‌های الگوریتم در زود همگرا شدن به دلیل افتادن در بهینگی محلی، از تکنیک فاز شروع دوباره استفاده شده است. همچنین، با تنظیم پویای پارامترهای الگوریتم از تکرارهای ابتدایی تا تکرارهای انتهایی ساختار جدیدی در الگوریتم ایجاد می‌شود که ابتدا تنوع جستجو زیاد و شدت بخشی جستجو کم است تا کل فضای حل مسأله قبل از رسیدن به بهینگی محلی یا سراسری بررسی شود تا مناطقی از فضای حل مسأله که دارای مقدار برازش هدف بهتری است مشخص شود. در تکرارهای بعدی الگوریتم، تنوع جستجو کم و شدت بخشی جستجو زیاد می‌شود تا الگوریتم در نواحی که از تکرارهای قبل شناسایی می‌کند از تکنیک‌های جستجوی محلی استفاده کند. با اجرای آزمایش روی ۹ تابع هدف پیوسته و مقایسه نتایج حاصل از الگوریتم پیشنهادی با نتایج روش‌های جستجوی هماهنگی ساده، جستجوی هماهنگی بهبود یافته و جستجوی هماهنگی بهترین کلی عملکرد مناسب الگوریتم پیشنهادی در اکثر نمونه‌ها نشان داده شد. همچنین از روش پیشنهادی برای حل مسأله زمان‌بندی استفاده شد که مقایسه نتایج

1. Largest Position Value

این روش و روش جستجوی هماهنگی ساده عملکرد موثر روش جدید را نشان می‌دهد. به منظور تحقیقات بیشتر می‌توان در سایر مسائل بهینه سازی در زمینه سیستم‌های تولیدی از روش پیشنهادی ارائه شده در این مقاله استفاده کرد.

مراجع

[1] Geem, Z.W., Kim J.H. and Loganathan G.V. (2001).A new heuristic optimization algorithm: harmony search. Simulations. 76, 60–68.

[2] Mönch, L., et al. (2011). Survey of problems, solution techniques, and future challenges in scheduling semiconductor-manufacturing operations. Journal of Scheduling. 14, 583-599.

[3] Yeniay, O. (2005). A comparative study on optimization methods for the constrained nonlinear programming problems. Mathematical Problems in Engineering. 2, 165–173.

[4] Lee, K. S., Geem Z.W., Lee S. H., Bae K. W. (2005). The harmony search heuristic algorithm for discrete structural optimization, Eng. Optim. 37, 663–684.

[5] Fesanghary, M., Mahdavi, M., Minary-Jolandan, M., Alizadeh,Y. (2008). Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Computer Methods in Applied Mechanics and Engineering. 197, 3080–3091.

[6] Espinoza, F.P., Minsker, B.S., Goldberg, D.E. (2001). A self-adaptive hybrid genetic algorithm. In: Proceedings on the Genetic and Evolutionary Computation Conference. San Francisco.

[7] Zahara, E., Hu, C.H. (2008). Solving constrained optimization problems with hybrid particle swarm optimization. Engineering Optimization. 40, 1031–1049.

[8] Pedamallu, C.S., Ozdamar, L. (2008). Investigating a hybrid simulated annealing and local search algorithm for constrained optimization, European Journal of Operations Research. 185, 1230–1245.

[9] Chen, X., Yang, J., Li, Z., Tian, D., Shao, Z. (2008). A combined global and local search method to deal with constrained optimization for continuous tabu search, International Journal for Numerical Methods in Engineering. 76, 1869–1891.

[10] Mahdavi, M., Fesanghary M. and Damangir E. (2007). An improved harmony search algorithm for solving optimization problems. Appl Math Comput. 188, 1567–1579.

[11] Omran, M.G.H., and Mahdavi. M. (2008). Global-best harmony search. Appl. Math. Comput. 198, 643–656.

[12] Pan, Q. K., Suganthan P.N., Fatih Tasgetiren M., Liang J.J. (2010). A self-adaptive global best harmony search algorithm for continuous optimization problems. Applied Mathematics and Computation. 216, 830–848.

[13] Zou, d., Gao, L., Steven L., Jianhua, W. (2011). Solving 0–1 knapsack problem by a novel global harmony search algorithm. Applied Soft Computing. 11, 1556–1564.

[14] Taherinejad, N. (2009). Highly Reliable Harmony Search Algorithm. IEEE Proceeding of European Conference on Circuit Theory and Design. pp. 818-822.

[15] Yao, X., Liu, Y., Lin, G. (1999). Evolutionary programming made faster. IEEE Transactions on Evolution Computation. 3, 82–102.